

Lösningarna till tentamensuppgifterna sätts ut på kurssidan på nätet under eftermiddagen.

Omtentamen i Programmeringsmetodik, 7,5 hp (DT2006, DT2010), 2008-01-12.

Hjälpmedel : Inga
Tid : 08:00-11:00
Ansvarig lärare : Christer Lindkvist 303393, 070-3273393

Svar till samtliga uppgifter 1-9 ska skrivas på utdelat extra papper. Använd ett papper till uppgifterna 1-3, två papper till uppgifterna 4-6 och ett papper per uppgift till uppgifterna 7-9. Skriv din tentamenskod på varje inlämnat extra papper.

Den maximala poängen för respektive uppgift står angiven i högermarginalen. Totalt kan 24 poäng erhållas. För betyget 3 krävs ca 12, för betyget 4 ca 17 och för betyget 5 ca 21,5 poäng.

Om inget speciellt anges gäller frågorna Visual C++.

Detta häfte ska du behålla.

Lycka till!

-
- 1) Antag att du har variablerna

```
int *ip, i = 5;
```

Visa hur man sätter **ip** att peka på **i** och hur man med **ip** dubblar värdet på **i**.

(1p)

- 2) Rita det binära träd som skapas då bokstäverna i ordet "mississippi" instoppas bokstav för bokstav i trädet i den angivna ordningen, om instoppningsfunktionen sätter in data som är mindre i högerträdet och lika stora eller större i vänsterträdet.

(1p)

- 3) Skriv de satser som stoppar in länken som **temp** pekar på mellan de två första (vänstra) länkarna i en envägslista enligt figur. Listan ska hänga ihop efteråt och pekaren **lista** ska peka på samma länk som tidigare.

(1p)

- 4) Skriv en **rekursiv** funktion, Upphojt(), som beräknar **ab** där a och b är heltal och ≥ 0 . Här följer en **rekursivt** definition:

(2p)

$ab = 1a \cdot ab - 1$ om $b = 0$ om $b > 0$

- 5) En lagervara kan avbildas som en abstrakt datatyp enligt:

```

/* Specifikation av vara -- vara.h */

#ifndef VARA_H
#define VARA_H

typedef struct {
    char artnr[11]; /* Varans artikelnummer */
    char namn[21]; /* Varans namn */
    int antal;     /* Antal i lager */
} vara;

void las_vara(vara *vp);
/* Frågar efter och läser in vara */

void skriv_vara(vara v);
/* Skriver ut vara på skärmen */

void salj_vara(vara *vp, int nr);
/* Sälj nr st av varan */

void fyll_vara(vara *vp, int nr);
/* Fyller på nr st av varan */

int antal_vara(vara vp);
/* Returnerar varans antal */

int lagre_artnr_vara(vara v1, vara v2);
/* Returnerar 1 om v1 har lägre artikelnummer än v2, annars 0 */

#endif

```

Implementera funktionerna **las_vara** enl. ovan. Namnet kan innehålla flera ord. För att rensa bufferten till och med nästa radslut kan funktionen **clr_line** användas:

(2p)

```

void clr_line(void) // Rensar resten av raden

```

- 6) Fullborda funktionen **rshift4**, så att den skiftar 32-bitarstalet **tal** fyra bitar åt höger (aritmetiskt skift) och returnerar det heltal som finns i de fyra minst signifikanta bitarna som skiftades ut. Funktionshuvud enligt:

(2p)

```

int rshift4(int *tal)

```

- 7) Skriv ett program som dynamiskt allokerar en vektor som ska innehålla ett inläst antal varor enligt uppgift 5 ovan. Läs först in antalet varor vektorn ska innehålla. Läs därefter in varorna till vektorn. Sortera vektorn m.a.p. artikelnummer med **bubbsort** (se sort.h nedan). Skriv slutligen ut varorna i

vektorn till binärfilen **Varor.dat**.

(5p)

```
/* Specifikation av sorteringsfunktioner -- sort.h */  
  
#include "vara.h"  
typedef vara datatyp;  
  
void bubbsort(datatyp v[], int nr, int (*comp)(datatyp, datatyp));  
/* Sorterar nr st element i v med bubbbelsortering */
```


-
- 8) Filen Krypto.txt är krypterad. Till varje tecken i filen har adderats ett pseudo
- 9) slumpstal (`rand() % 32`). Dessutom är filen lagrad baklänges, så att sista tecknet kommer först. Skriv ett fullständigt program som dekrypterar filen och sparar resultatet i textfilen Plain.txt. Vid dekrypteringen läses filen tecken för tecken samtidigt som tecknen lägges på en stack. Därefter initieras slumpstalsgeneratoren med 9999. Sedan hämtas tecknen från stacken, ett slumpstal (`rand() % 32`) subtraheras från respektive tecken som därefter skrivs till filen Plain.txt. För hantering av stacken ska du använda:

(5p)

```

/* Specifikation av LIFO-lista -- lifo.h */

#ifndef LIFOH
#define LIFOH

typedef char datatyp;      /* Exempelvis */

typedef struct link {
    datatyp data;
    struct link *next;
} linktyp;

void push(linktyp **lpp, datatyp d);
/* Stoppar in d i LIFO-listan */

datatyp pop(linktyp **lpp);
/* Tar bort data från LIFO-listan */

#endif

```

- 10) I binärfilen **Varor.dat** finns ett antal lagervaror. Skriv ett program som läser hela filen och stoppar in varorna i en tvåvägslista. Listan ska vara sorterad efter artikel
- 11) nummer. Avslutningsvis ska programmet skriva ut alla varor som har negativt lagersaldo (antal) på skärmen.

För hantering av lagervaror ska du använda den abstrakta datatypen **vara** i uppgift 5 och för hantering av tvåvägslistorna ska du använda den abstrakta data

typen **twolist** nedan.

(5p)

```

/* Specifikation av tvåvägslista -- twolist.h */

#ifndef TWOLISTH

```



```

#define TWOLISTH

#include "medlem.h"
typedef medlem datatyp;

typedef struct twolink {
    enum {head, link} kind;
    struct twolink *befo, *next;
    datatyp data;
} headtyp, linktyp;

void newhead(headtyp **hpp);
/* Skapar en ny tom lista */

void newlink(linktyp **lpp);
/* Skapar en ny tom länk */

void putlink(datatyp d, linktyp *lp);
/* Sätter in data i en länk */

datatyp getlink(linktyp *lp);
/* Returnerar data från länk */

void inlast(linktyp *lp, headtyp *hp);
/* Sätter in länken sist i listan */

void infirst(linktyp *lp, headtyp *hp);
/* Sätter in länken först i listan */

void inpred(linktyp *lp, linktyp *ep);
/* Sätter in första länken före den andra */

void insucc(linktyp *lp, linktyp *ep);
/* Sätter in första länken efter den andra */

void insort(linktyp *lp, headtyp *hp,
            int (*is_less)(datatyp d1, datatyp d2));
/* Sätter in länken sorterad enligt is_less */

linktyp *firstlink(headtyp *hp);
/* Returnerar pekare till första länken i listan */

linktyp *lastlink(headtyp *hp);
/* Returnerar pekare till sista länken i listan */

linktyp *predlink(linktyp *lp);
/* Returnerar pekare till länken före */

linktyp *succlink(linktyp *lp);
/* Returnerar pekare till länken efter */

int is_link(linktyp *lp);
/* Returnerar 1 om länk annars 0 */

int empty(headtyp *hp);
/* Returnerar 1 om listan tom annars 0 */

int nrlinks(headtyp *hp);
/* Returnerar antalet länkar i listan */

void outlist(linktyp *lp);
/* Tar bort länken från listan */

```

```
void elimlink(linktyp **lpp);  
/* Tar bort, avallokerar och NULL-ställer länken */  
  
void clearhead(headtyp *hp);  
/* Tar bort alla länkar från listan */  
  
void elimhead(headtyp **hpp);  
/* Eliminerar och NULL-ställer listan */  
  
#endif
```

Lösningar till tentamen i Programmeringsmetodik, 5p, 2008-01-12

- 1)

```
ip = &i;
*ip = *ip * 2;    // Alternativ: *ip *= 2;
```
- 2)
- 3)

```
temp->next = lista->next;
lista->next = temp;
```
- 4)

```
int upphojt(int a, int b)
{
    if (b == 0)
        return 1;
    else
        return a*upphojt(a, b-1);
}
```
- 5)

```
#include <stdio.h>
#include <string.h>
#include "vara.h"

void las_vara(vara *vp)
{
    printf("Artikelnr? ");
    scanf("%10s", vp->artnr);
    clr_line();
    printf("Namn? ");
    fgets(vp->namn, sizeof(vp->namn), stdin);
    if (vp->namn[strlen(vp->namn)-1] == '\n')
        vp->namn[strlen(vp->namn)-1] = '\0';    // Trimma bort '\n'
    else
        clr_line();
    printf("Antal? ");
    scanf("%d", &vp->antal);
    clr_line();
}
```

```

6)  int rshift4(int *tal)
    {
        int spill = *tal & 0xF;

        if (*tal < 0)
            *tal = (*tal >> 4) | 0xF0000000;
        else
            *tal = (*tal >> 4) & 0x0FFFFFFF;
        return spill;
    }

7)  #include <stdio.h>
    #include <stdlib.h>
    #include "vara.h"
    #include "sort.h"

    void main()
    {
        FILE *bsut;
        int i, nr;
        vara *vp;

        printf("Ge antal varor: ");
        scanf("%d", &nr);
        vp = calloc(nr, sizeof(vara));
        for (i = 0; i < nr; i++) {
            las_vara(&vp[i]);
        }

        bubbsort(vp, nr, lagre_artnr_vara);

        bsut = fopen("varor.dat", "wb");
        for (i = 0; i < nr; i++) {
            skriv_vara(vp[i]);
            fwrite(&vp[i], sizeof(vara), 1, bsut);
        }
        fclose(bsut);

        free(vp);
    }

```

8)

```

9) #include <stdio.h>
#include <stdlib.h>
#include "lifo.h"

void main()
{
    FILE *tsin, *tsut;
    linktyp *lp = NULL;
    char ch;

    tsin = fopen("Krypto.txt", "rt");
    while (fscanf(tsin, "%c", &ch) != EOF) {
        push(&lp, ch);
    }
    fclose(tsin);

    srand(9999);
    tsut = fopen("Plain.txt", "wt");
    while (lp != NULL) {
        ch = pop(&lp) - rand() % 32;
        fprintf(tsut, "%c", ch);
    }
    fclose(tsut);
}

10) #include <stdio.h>
#include "vara.h"
#include "twolist.h"

void main()
{
    FILE *bsin;
    headtyp *hp;
    linktyp *lp;
    vara v;

    /* Läs data från fil till lista */
    newhead(&hp);
    bsin = fopen("varor.dat", "rb");
    fread(&v, sizeof(v), 1, bsin);
    while (!feof(bsin)) {
        newlink(&lp);
        putlink(v, lp);
        insort(lp, hp, lagre_artnr_vara);
        fread(&v, sizeof(v), 1, bsin);
    }
    fclose(bsin);

    /* Skriv ut varor med negativt lagersaldo */
    lp = firstlink(hp);
    while (lp != NULL) {
        v = getlink(lp);
        if (antal_vara(v) < 0)
            skriv_vara(v);
        lp = succlink(lp);
    }
    elimhead(&hp);
}

```